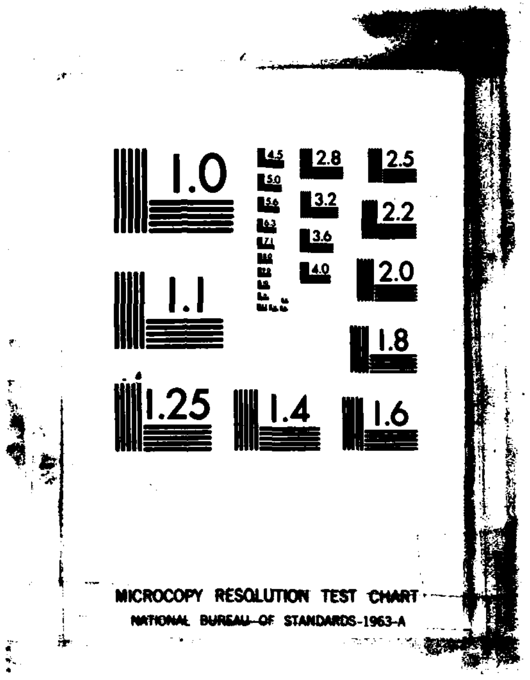
AD-A141 186   NNTERACTIVE PROGRAMMING(U) STANFORD UNIV CA DEPT OF COMPUTER SCIENCE Z MANNA DEC 83 AFOSR-TR-84-0297 AFOSR-81-0014   1/1

UNCLASSIFIED   F/G 9/2   NL

END


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
7.1
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

INTERACTIVE PROGRAMMING

by

Zohar Manna
Professor of Computer Science
Stanford University Stanford, CA 94305

Interim Scientific Report:
Air Force Office of Scientific Research
Grant AFOSR-81-0014
Oct. 1, 1982 - Sept. 30, 1983

Our research was concentrated on the following topics:

1. Verification of Concurrent Programs: A Proof System ([1]).

A proof system based on temporal logic is presented for proving properties of concurrent programs. The system consists of three parts: the general uninterpreted part, the domain-dependent part and the program-dependent part. In the general part we give a complete system for first-order temporal logic with detailed proofs of useful theorems. This logic enables reasoning about general time sequences. The domain-dependent part characterizes the special properties of the domain over which the program operates. The program-dependent part introduces program axioms which restrict the time sequences considered to be execution sequences of a given program.

The utility of the full system is demonstrated by proving invariance, liveness and precedence properties of several concurrent programs. Derived proof principles for these classes of properties are obtained which lead to compact representation of proofs.

2. Temporal Proof System for General Languages ([2]).

An abstract temporal proof system is presented whose program-dependent part has a high-level interface with the programming language actually studied. Given a new language, it is sufficient to define the interface notions of atomic transitions, justice, and fairness in order to obtain a full temporal proof system for this language. This construction is particularly useful for the analysis of concurrent systems. We illustrate the construction on the shared-variable model and on CSP. The generic proof system is shown to be relatively complete with respect to pure first-order temporal logic.

3. Proving Precedence Properties: The Temporal Way ([3])

We explore the three important classes of temporal properties of concurrent programs: invariance, liveness and precedence. We present the first methodological approach to the precedence properties, while providing a review of the invariance and liveness properties. The approach is based on the *unless* operator $\mathcal{U}$ which is a weak version of the *until* operator $\mathcal{U}$. For each class of properties, we present a single complete principle. Finally, we show that the properties of each class are decidable over finite state programs.

4. Verification of Concurrent Programs: Adequate Proof Principles ([4]).

We present proof principles for establishing invariance, eventuality and until properties. The methods for liveness are based on *well-founded assertions* and are applicable to both "just" and "fair" computations. These methods do not assume a decrease of the rank at each computation step. It is sufficient that there exists one process which decreases the rank when activated. Fairness then ensures that the program will eventually attain its goal. In the finite state case the proofs can be represented by diagrams.

5. Synthesis of Communicating Processes from Temporal Specifications ([5],[6]).

We apply Propositional Temporal Logic (PTL) to the specification and synthesis of the synchronization part of communicating processes. To specify a process, we give a PTL formula that describes its sequence of communications. The synthesis is done by constructing a model of the given specifications using a tableau-like satisfiability algorithm for PTL. This model can then be interpreted as a program.

6. Special Relations in Program Synthetic Deduction ([7]).

Program synthesis is the automated derivation of a computer program from a given specification. In the *deductive approach*, the synthesis of a program is regarded as a theorem-proving problem; the desired program is constructed as a by-product of the proof. We present a formal deduction system for program synthesis, with special features for handling equality, the equivalence connective, and ordering relations.

In proving theorems involving the equivalence connective, it is awkward to remove all the quantifiers before attempting the proof. The system therefore deals with *partially skolemized sentences*, in which some of the quantifiers may be left in place. A rule is provided for removing individual quantifiers when required after the proof is under way.

The system is also *nonclausal*; i.e., the theorem does not need to be put into conjunctive normal form. The equivalence, implication, and other connectives may be left intact.

7. The Logical Basis for Computer Programming ([8]).

This is an introductory textbook, divided into two volumes.

The first volume, subtitled *Deductive Reasoning*, describes several logical structures and presents methods for the informal but rigorous proof of theorems (or properties) about these structures. In this volume, we introduce the basic notions of propositional and predicate logic, and theories with equality and with mathematical induction. We describe within theories with induction some of the most important structures of computer science, including the integers, strings, trees, lists, sets, tuples (arrays), and sequences. We apply logical methods to establish in these theories properties such as the correctness of algorithms for parsing (of strings) and sorting (of tuples). The induction principles of the various theories are then unified into a single well-founded induction principle.

The second volume, subtitled *Deductive Techniques*, presents methods for the formal proof of such theorems, oriented toward the development of computer theorem-proving systems. In this second volume, we apply the concepts of the first volume to develop more formal proof techniques. We first describe an additional theory with induction, the theory of expressions and substitutions. Within this theory, we describe the unification algorithm and prove its correctness. We then introduce special logical techniques essential in theorem-proving systems, such as skolemization and polarity. We present a deductive system for describing formal proofs; this framework incorporates the most useful logical techniques for theorem proving, including resolution, rewriting rules, and proof by mathematical induction.

8. Reasoning About Digital Circuits ([9], [10]).

We present a formalism called *interval temporal logic* (ITL) that augments standard predicate logic with time-dependent operators. ITL is like discrete linear-time temporal logic but can describe time intervals. The behavior of programs and hardware devices can often be decomposed into successively smaller intervals of activity. State transitions can be characterized by properties relating the initial and final values of variables over intervals. Furthermore, these time periods provide a convenient framework for introducing quantitative timing details.

We presented the propositional and first-order syntax and semantics of ITL. We demonstrate ITL's utility for uniformly describing the structure and dynamics of a wide variety of timing-dependent digital circuits. Devices considered include delay elements, adders, latches, flip-flops, counters, random-access memories, a clocked multiplication circuit and the Am2901 bit slice. ITL also provides a means for expressing properties of such specifications. We examined such concepts as device equivalence and internal states. Propositional ITL was shown to be undecidable although useful subsets are of relatively reasonable computational complexity.

Publications

[1] Z. Manna, A. Pnueli, "Verification of Concurrent Programs: a Temporal Proof System," Proc. 4th School on Advanced Programming, Amsterdam, Holland (June 1982), pp. 163-255.

[2] Z. Manna, A. Pnueli, "How to Cook a Temporal Proof System for Your Pet Language," Proc. of the Symposium on Principles of Programming Languages, Austin, Texas (Jan. 1983), pp. 141-154

[3] Z. Manna, A. Pnueli, "Proving Precedence Properties – the Temporal Way," ICALP Conference, Barcelona, Spain (July 1983).

[4] Z. Manna, A. Pnueli, "Adequate Proof Principles for Invariance and Liveness Properties of Concurrent Programs," Technical Report, Computer Science Department, Stanford University, 1982 (to appear in TOPLAS).

[5] Z. Manna, P. Wolper, "Synthesis of Communicating Processes from Temporal Specifications," Proc. of the Workshop on Logics of Programs (Yorktown Heights), N.Y., Springer-Verlag Lecture Notes in Computer Science, 1981 (to appear also in the ACM Trans. on Programming Languages and Systems, 1983).

[6] P. Wolper (supervised by Z. Manna), "Synthesis of Communicating Processes from Temporal Logic Specifications," Ph.D. Thesis, Computer Science Dept., Stanford University (August 1982).

[7] Z. Manna, R. Waldinger, "Special Relations in Program Synthetic Deductions," Journal of the ACM (to appear, 1983).

[8] Z. Manna, R. Waldinger, "The Logical Basis for Computer Programming," 2 volumes, in preparation.

[9] Z. Manna, B. Moszkowski, "A Hardware Semantics Based on Temporal Intervals," ICALP Conference, Barcelona, Spain (July 1983).

[10] B. Moszkowski (supervised by Z. Manna), "Reasoning about Digital Circuits," Ph.D. Thesis, Computer Science Department, Stanford University (July 1983).

INTERACTIVE PROGRAMMING

by

Zohar Manna
Professor of Computer Science
Stanford University Stanford, CA 94305

Interim Report of Inventions:
Air Force Office of Scientific Research
Grant AFOSR-81-0014
Oct. 1, 1982 - Sept. 30, 1983

There are no patents or copyrights to report.

[illegible]

Zohar Manna

6a. NAME OF PERFORMING ORGANIZATION Stanford University	**6b. OFFICE SYMBOL** *(If applicable)*	**7a. NAME OF MONITORING ORGANIZATION** Air Force Office of Scientific Research
6c. ADDRESS *(City, State and ZIP Code)* Department of Computer Science Stanford CA · 94305		**7b. ADDRESS** *(City, State and ZIP Code)* Directorate of Mathematical & Information Sciences, Bolling AFB DC 20332
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AFOSR	**8b. OFFICE SYMBOL** *(If applicable)* NM	**9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER** AFOSR-81-0014

8c. ADDRESS *(City, State and ZIP Code)* Bolling AFB DC 20332	**10. SOURCE OF FUNDING NOS.** PROGRAM ELEMENT NO. 61102F	PROJECT NO. 2304	TASK NO. A4	WORK UNIT NO.
11. TITLE *(Include Security Classification)* INTERACTIVE PROGRAMMING				

12. PERSONAL AUTHOR(S) Zohar Manna

13a. TYPE OF REPORT Interim	**13b. TIME COVERED** FROM 1/10/82 TO 30/9/83	**14. DATE OF REPORT** *(Yr., Mo., Day)* DEC 83	**15. PAGE COUNT** 5

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
FIELD	GROUP	SUB. GR.	

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

During this period, research was concentrated on the following topics: (1) Verification of concurrent programs: a proof system - A proof system based on temporal logic is presented for proving properties of concurrent programs. (2) Temporal proof system for general languages - An abstract temporal proof system is presented whose program-dependent part has a high-level interface with the programming language actually studied. (3) Proving precedence properties - the temporal way - The investigators explore the three important classes of temporal properties of concurrent programs: invariance, liveness and precedence. (4) Verification of concurrent programs - adequate proof principles - The investigators ~~present proof~~ principles for establishing invariance, eventuality and until properties. (5) Synthesis of communicating processes from temporal specifications - The investigators apply Propositional Temporal Logic (PTL) to the specification and synthesis of the synchronization part of communicating processes. (CONTINUED)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	**21. ABSTRACT SECURITY CLASSIFICATION** UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. Robert N. Buchal	**22b. TELEPHONE NUMBER** *(Include Area Code)* (202) 767- 4939	**22c. OFFICE SYMBOL** NM

ITEM #19, ABSTRACT, CONTINUED: (6) Special relations in program synthetic deduction - Program synthesis is the automated derivation of a computer program from a given specification. (7) The logical basis for computer programming - This is an introductory textbook, divided into two volumes; the first volume, subtitled Deductive Reasoning, describes several logical structures and presents methods for the informal but rigorous proof of theorems (or properties) about these structures. The second volume, subtitled Deductive Techniques, presents methods for the formal proof of such theorems, oriented toward the development of computer theorem-proving systems. (8) Reasoning about digital circuits - The investigators present a formalism called 'interval temporal logic' (ITL) that augments standard predicate logic with time-dependent operators.

FILM